ИНФОРМАТИКА, ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА И УПРАВЛЕНИЕ
INFORMATION TECHNOLOGY, COMPUTER SCIENCE, AND MANAGEMENT



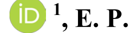


УДК 004.722:519.172 https://doi.org/10.23947/2687-1653-2021-21-3-284-289

Метод формирования графа локальной сети на основе анализа множеств адресов






В. В. Галушка ✉[1], **Д. В. Фатхи** [1], **Е. Р. Газизов** [2]

[1] ФГБОУ ВО «Донской государственный технический университет» (г. Ростов-на-Дону, Российская Федерация)

[2] ФГБОУ ВО «Казанский государственный аграрный университет» (г. Казань, Российская Федерация)

✉ galushkavv@yandex.ru

Введение. Статья посвящена вопросам автоматизированного построения схемы локальной вычислительной сети с использованием средств и методов анализа трафика на канальном уровне модели OSI. Проблема обусловлена двумя факторами. Это сложности ручного определения связей между оборудованием и отсутствие физического доступа к линиям связи уже функционирующей сети. Цель работы — сокращение времени, затрачиваемого на построение схемы локальной сети, за счет автоматизации процесса определения связей между оборудованием.

Материалы и методы. Для решения поставленных задач предложен метод определения взаимного расположения устройств. Задействованы направленные в противоположные стороны сетевые адаптеры специализированного программно-аппаратного комплекса, подключаемого в разрыв линии связи в разных точках сети. Используемый метод базируется на вычислениях пересечений множеств адресов, полученных с этих адаптеров. Приведены структурные схемы построения такого программно-аппаратного комплекса и требования к нему. Описаны способы получения MAC-адресов из транзитных пакетов. Приводятся примеры библиотек программных компонентов для выполнения этой операции. Для хранения полученных данных предложена структура реляционной базы данных. Описаны формат и содержание полей ее таблицы.

Результаты исследования. С использованием разработанных методов на типовом примере сети стандарта Ethernet показан способ определения взаимного расположения конечных устройств, заданных своими MAC-адресами, а также как минимум двух коммутаторов, находящихся между ними. Определены признаки, по которому можно судить о наличии коммутационного оборудования в том или ином сегменте. Предложен метод, позволяющий с использованием набора реляционных операций последовательно уточнять топологию сети до достижения требуемой точности.

Обсуждение и заключения. Полученные результаты могут быть использованы при администрировании крупных локальных сетей с разветвленной структурой. Предложенный подход позволяет сократить время на построение схемы. Это возможно благодаря автоматизации процесса получения информации о работающих в сети устройствах и их взаимном расположении.

Ключевые слова: топология сети, граф, дерево, локальная сеть, анализ трафика, множества, реляционные операции.

Для цитирования: Галушка, В. В. Метод формирования графа локальной сети на основе анализа множеств адресов / В. В. Галушка, Д. В. Фатхи, Е. Р. Газизов // Advanced Engineering Research. — 2021. — Т. 21, № 3. — С. 284–289. https://doi.org/10.23947/2687-1653-2021-21-3-284-289

A method for generating a local network graph based on the analysis of address sets

V. V. Galushka ✉[1], D. V. Fatkhi [1], E. R. Gazizov [2]
[1] Don State Technical University (Rostov-on-Don, Russian Federation)
[2] Kazan Agricultural State University (Kazan, Russian Federation)
✉ galushkavv@yandex.ru

Introduction. The paper deals with the problem of automated construction of a local area network using tools and methods for traffic analysis at the link layer of OSI model. The problem is caused by two factors. These are difficulties of the manual determination of the communication between equipment and the lack of physical access to communication lines of an already functioning network. The purpose of the work is to reduce the time spent on building a local network diagram through automating the process of determining the communication between the equipment.

Materials and Methods. To solve the set tasks, a method for determining the relative location of devices is proposed. The network adapters of a specialized software and hardware complex, which are connected to a communication line break at different points of the network, are used in opposite directions. The method used is based on calculations of intersections of address sets received from these adapters. The structural schemes of the construction of such a software and hardware complex and the requirements for it are given. The methods of obtaining MAC addresses from transit packets are described. Examples of libraries of software components for performing this operation are given. The structure of a relational database is proposed for storing the received data. The format and content of the fields of its table are described.

Results. Using the developed methods, a typical example of an Ethernet network shows a way to determine the relative location of end devices specified by their MAC addresses, as well as at least two switches located between them. The signs by which it is possible to judge the presence of switching equipment in a particular segment are determined. A method is proposed that enables through using a set of relational operations, to sequentially refine the network topology until the required accuracy is achieved.

Discussion and Conclusions. The results obtained can be used under the administration of large local networks with an extensive structure. The proposed approach allows you to reduce the time required for building a scheme. This is possible due to the automation of the process of obtaining information about devices operating on the network and their mutual location.

Keywords: network topology, graph, tree, local network, traffic analysis, sets, relational operations.

For citation: V. V. Galushka, D. V. Fatkhi, E. R. Gazizov. A method for generating a local network graph based on the analysis of address sets. Advanced Engineering Research, 2021, vol. 21, no. 3, pp. 284–289. https://doi.org/10.23947/2687-1653-2021-21-3-284-289

Введение. Для крупных локальных вычислительных сетей характерна сложная конфигурация физических связей, которая во многом определяет эффективность их работы [1, 2]. На практике в организации далеко не всегда есть подробная схема или иная документация, описывающая сетевое оборудование и связи между ним. Это значительно осложняет процедуры администрирования и обусловливает актуальность проблемы определения структуры связей в эксплуатируемой сети для дальнейшего построения схемы расположения и соединения узлов.

Линии связи чаще всего скрыты за элементами конструкции или отделки здания, доступно только коммутационное оборудование. В таком случае невозможно понять, к какому из узлов сети ведет каждый подключенный кабель. Поэтому возникает задача построения схемы сети на основе анализа данных, полученных из трафика, захваченного в определенных точках. Речь идет о местах, которые потенциально доступны для подключения дополнительных программно-аппаратных средств, анализирующих трафик. Цель — сокращение времени, затрачиваемого на построение схемы локальной сети.

Описанная задача осложняется тем, что вся информация, касающаяся функционирования локальной сети, относится ко второму (канальному) уровню модели OSI, а значительная часть важных данных в пакете относится к более высокому уровню — сетевому [3, 4]. Большинство способов анализа трафика рассчитаны на обработку информации сетевого уровня [5]. В связи с этим возникает необходимость разработки методов, позволяющих получить все нужные данные для построения схемы сети из заголовков пакетов канального уровня модели OSI. С другой стороны, топологии сетей на канальном уровне проще, чем на сетевом, и всегда строго регламентируются соответствующими стандартами[1].

[1] IEEE 802.3-2018 — IEEE Standard for Ethernet / IEEE Standard Association // standards.ieee.org/ : [сайт]. — URL: https://standards.ieee.org/standard/802_3-2018.html (дата обращения: 11.04.2021).

Материалы и методы. Стандарт Ethernet, повсеместно используемый для построения локальных компьютерных сетей, предусматривает использование топологии «дерево» для организации связей между узлами [5]. В теории графов «дерево» определяется как связный граф без циклов [6]. Важное следствие из этого определения: между любыми парами вершин в дереве имеется один и только один путь [7]. Это позволяет отказаться от поиска маршрутов в пределах такой сети и значительно упростить работу оборудования.

При построении графа определяется множество его вершин и связей между ними [8]. Применительно к графу сети вершины — это сетевое оборудование. Для его адресации в пределах локальной сети используются присвоенные производителем MAC-адреса. Они уникальны для каждого устройства и имеют размер 6 байтов [9]. В заголовке каждого сетевого пакета — два MAC-адреса: отправителя и получателя. Они не меняются во время передачи пакета в пределах локальной сети и поэтому в рассматриваемой задаче могут быть использованы для идентификации узлов сети.

При построении графа сети основная сложность — определение связей. Каждая вязь соединяет две вершины, взаимное расположение которых, как отмечалось выше, неизвестно из-за их большой удаленности или скрытой прокладки телекоммуникаций. Связи могут соединять устройства разных типов: коммутатор — компьютер или коммутатор — коммутатор. Последние составляют инфраструктуру передачи данных и представляют наибольший интерес с точки зрения анализа топологии сети. В отличие от них соединения коммутационного оборудования с компьютерами описывают конечные вершины графа. При этом компьютеры, подключенные к одному коммутатору, можно условно объединить в группу, так как для них взаимное расположение относительно других компьютеров будет одинаковым. В качестве группы можно рассматривать и более крупные множества узлов, в том числе компьютеры, подключенные к двум или нескольким ближайшим коммутаторам (например, отвечающим за связь в пределах одного этажа здания или нескольких кабинетов одного отдела). В целом узлы, входящие в множество, должны располагаться друг к другу ближе, чем к узлам, не входящим в множество или входящим в другое множество [10]. В первом приближении всю локальную сеть можно рассматривать как такое множество, потому что ее узлы тесно связаны между собой и отделены от других сетей [11].

Идея исследования состоит в последовательном уточнении топологии сети. Для этого разделим множество MAC-адресов входящих в нее устройств на более мелкие подмножества вплоть до определения групп компьютеров, подключенных к отдельным коммутаторам.

Разделение на подмножества выполняется относительно точек, в которых к сети подключается аппаратное устройство, способное анализировать сетевые пакеты и извлекать из них адресную и другую информацию. Таким устройством может быть ноутбук или одноплатный компьютер, поддерживающие работу одновременно с двумя сетевыми адаптерами. Это позволит подключать их в разрыв соединения. В результате к каждому из двух сетевых адаптеров будет подключена часть сети.

Важно отметить разницу между терминами «вершина» и «точка». Вершина — это часть графа сети, обозначающая какое-либо оборудование: коммутатор или конечное устройство. Точка — место подключения указанного аппаратного комплекса, которое всегда находится между двумя вершинами.

Учитывая подключение анализирующего устройства в разрыв соединения, необходимо обеспечить работоспособность той линии связи, в которой этот разрыв создан. Для этого сетевые адаптеры должны быть связаны средствами операционной системы. Используется соединение типа «мост», при котором пакеты, пришедшие на один из интерфейсов, передаются на другой при помощи механизмов канального уровня модели OSI, то есть без учета IP-адресов, маршрутизации, NAT и т. д. Такой способ организации соединения полностью прозрачен для других устройств в сети, он не изменяет пакеты и не проявляет себя каким-либо другим образом.

Основная задача рассматриваемого устройства — извлечение MAC-адресов из транзитных пакетов. На этом (первом) этапе построения графа локальной сети используются утилиты захвата трафика (его записывают в файл и анализируют) или специализированные библиотеки программных компонентов (анализируют трафик в реальном времени) [12, 13]. В зависимости от операционной системы библиотеки могут отличаться, однако, как правило, все они основаны на Pcap (Packet Capture).

Независимо от способа получения MAC-адресов, информация о них должна сохраняться в базе данных. Учитывая описанные ранее особенности процесса построения схемы сети, отметим следующие требования. Для каждого MAC-адреса записывается дополнительная информация:

— о точке, в которую подключено устройство, получившее MAC-адрес;

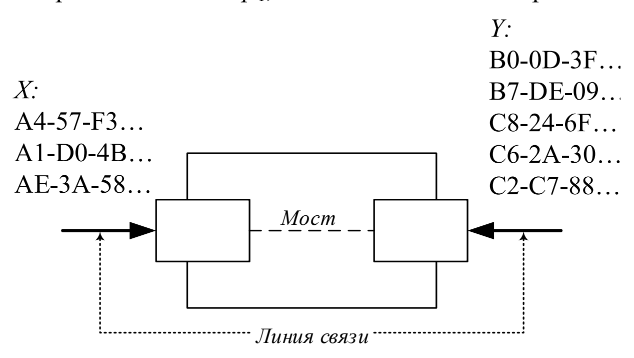
— о сетевом интерфейсе, с которого MAC-адрес получен как адрес отправителя [14].

В итоге таблица базы данных будет описываться отношением A со следующей схемой:

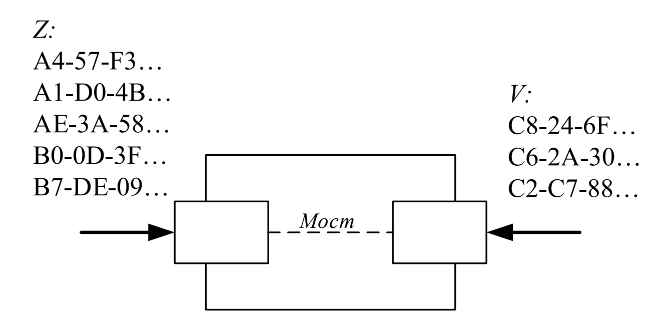
$$A\ (\underline{id},\ address,\ point,\ side).$$

Здесь *id* — первичный ключ, используемый только для идентификации записей в таблице; *address* — MAC-адрес устройства в сети, извлеченный из проходящего пакета; *point* — точка подключения в сети (физическое место); *side* — условное обозначение сетевого интерфейса, передавшего пакет, из которого извлекли MAC-адрес.

После формирования базы данных MAC-адресов для некоторого количества точек захвата трафика начинается следующий этап — построение схемы сети. Оно базируется на информации о распределении MAC-адресов, полученной для разных точек подключения. Обозначим произвольные две из них как p_1 и p_2. Для каждой точки должны быть получены два множества адресов, каждое — от отдельного сетевого адаптера. Обозначим X и Y — множества адресов для точки p_1, Z и V — множества адресов для точки p_2 (рис. 1).



Подключение в точке p_1



Подключение в точке p_2

Рис. 1. Распределение адресов по множествам при подключении в разные точки сети

Здесь и далее у MAC-адресов для сокращения записи указана только первая часть. В рамках рассматриваемого примера она уникальна, и этого достаточно для отражения работы метода.

На основе полученного распределения адресов по множествам, соответствующим разным сетевым интерфейсам, можно сделать первоначальные выводы о взаимном расположении устройств. Для этого необходимо вычислить все возможные пересечения для двух точек, то есть $X \cap Z$, $X \cap V$, $Y \cap Z$, $Y \cap V$.

Пересечения целесообразно вычислять средствами системы управления базами данных. Это обусловлено тем, что:

— информация о принадлежности адреса ко множеству хранится в базе данных,

— в реляционной алгебре поддерживаются операции над множествами [15].

Необходимо выполнить запросы, эквивалентные следующему набору выражений:

$$X \cap Z = \Pi_{address}\,(\sigma_{point\,=\,1\,\wedge\,side\,=\,1}(A)) \cap (\sigma_{point\,=\,2\,\wedge\,side\,=\,1}(A))),$$

$$X \cap V = \Pi_{address}\,(\sigma_{point\,=\,1\,\wedge\,side\,=\,1}(A)) \cap (\sigma_{point\,=\,2\,\wedge\,side\,=\,2}(A))),$$

$$Y \cap Z = \Pi_{address}\,(\sigma_{point\,=\,2\,\wedge\,side\,=\,1}(A)) \cap (\sigma_{point\,=\,1\,\wedge\,side\,=\,1}(A))),$$

$$Y \cap V = \Pi_{address}\,(\sigma_{point\,=\,2\,\wedge\,side\,=\,2}(A)) \cap (\sigma_{point\,=\,1\,\wedge\,side\,=\,2}(A))).$$

Результаты исследования. Рассмотрим пример применения предлагаемой методики для построения топологии сети, исходя из распределения множеств MAC-адресов, приведенного на рис. 1. Определим необходимые пересечения множеств:

$$X \cap Z = \{A4 - 57 - F3,\ A1 - D0 - 4B,\ AE - 3A - 58\},$$

$$X \cap V = \emptyset,$$

$$Y \cap Z = \{B0 - 0D - 3F,\ B7 - DE - 09\},$$

$$Y \cap V = \{C8 - 24 - 6F,\ C6 - 2A - 30,\ C2 - C7 - 88\}.$$

Можно заметить, что одно из пересечений (X и V) — пустое множество. Такой результат получается для противоположно направленных сторон. Соответственно, другие множества (Y и Z), наоборот, представляют направленные друг на друга стороны, а результат их пересечения — это адреса, находящиеся между точками измерения, то есть между p_2 и p_1.

Остальные пересечения представляют адреса, находящиеся по разные стороны за точками измерений. X и V представляют противоположно направленные стороны. Поэтому оставшееся пересечение, в котором участвует X (то есть $X \cap Z$), включает в себя адреса, находящиеся со стороны точки p_1, $Y \cap V$ — со стороны точки p_1. Таким образом, можно сделать первоначальный вывод о взаимном расположении всех полученных в ходе анализа адресов, а также об их расположении относительно точек измерения (рис. 2).

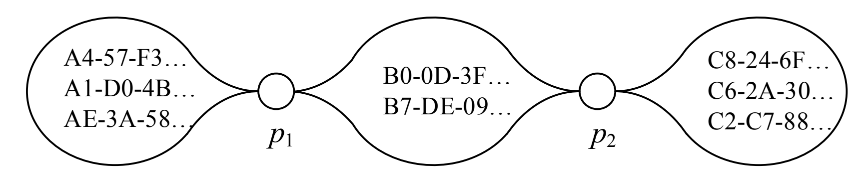


Рис. 2. Взаимное расположение устройств и точек

Следует помнить, что точки на данной схеме не являются узлами сети (в частности, коммутаторами). Однако полученные результаты позволяют сделать предположение: если в множество входят несколько адресов, значит, внутри него есть как минимум один коммутатор. Данное утверждение обоснуем так: несколько компьютеров не могут быть соединены напрямую, для этого требуется соответствующее сетевое оборудование (рис. 3).

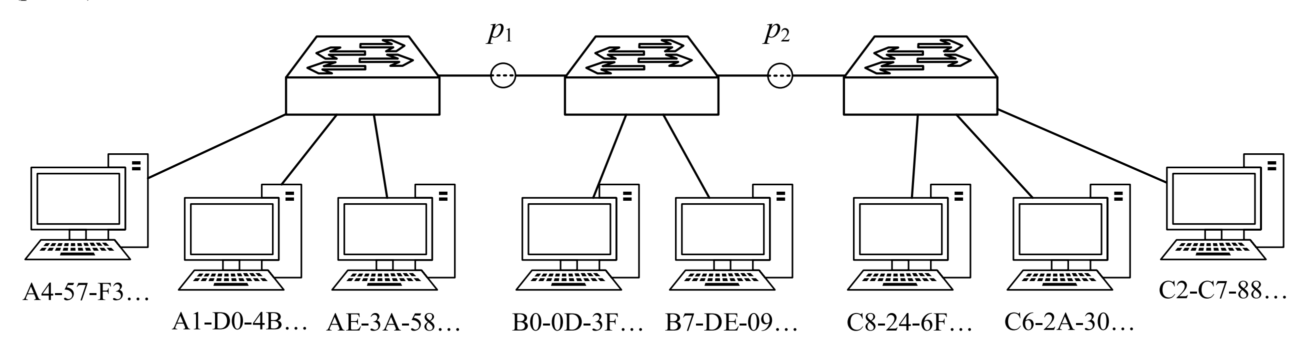


Рис. 3. Схема сети

Схема, представленная на рис. 3, не окончательная, так как внутри каждого из множеств может быть не один, а несколько коммутаторов. На следующих этапах работы метода для каждого из полученных множеств следует выполнить аналогичные операции, получив MAC-адреса в других точках сети. Каждое новое измерение позволит уточнить схему и дополнить ее новыми коммутационными узлами.

Обсуждение и заключения. Предложен способ эффективного построения схемы сети. Подход базируется на автоматизированном анализе открытой информации, извлеченной из передаваемых по сети пакетов. Данная методика — альтернатива физическому поиску линий связи и определению соединенных ими устройств. Применение предложенных решений позволит значительно сократить время, затрачиваемое системными администраторами на определение местоположения всех устройств и нанесение их на схему сети. Преимущество методики — возможность последовательного уточнения топологии сетевых связей до получения требуемой точности.

Библиографический список

1. Кузьменко, Н. Г. Компьютерные сети и сетевые технологии / Н. Г. Кузьменко. — СПб. : Наука и техника, 2013. — 368 с.

2. Галушка, В. В. Сети и системы передачи информации / В. В. Галушка. — Ростов-на-Дону : Изд. центр ДГТУ, 2016. — 105 с.

3. Orzen, S.-N. Interaction understanding in the OSI model functionality of networks with case studies / Stefano-Niko Orzen // IEEE 9th Int. SACI. — 2014. — P. 327–330. — URL: www.researchgate.net/publication/269301474_Interaction_understanding_in_the_OSI_model_functionality_of_networks_with_case_studies (accessed: 18.08.2021). 10.1109/SACI.2014.6840086

4. Saxena, P. OSI Reference Model — A Seven Layered Architecture of OSI Model / Piyush Saxena // International Journal of Research. — 2014. — Vol. 1 (10). — P. 1145–1156.

5. Лагутин, И. А. Определение топологии с помощью протокола LLDP в сетях Juniper / И. А. Лагутин // Перспективы развития информационных технологий : [сайт]. — 2013. — № 16. — С. 66–70. — URL: https://cyberleninka.ru/article/n/opredelenie-topologii-s-pomoschyu-protokola-lldp-v-setyah-juniper/viewer (дата обращения: 10.04.2021).

6. Алексеев, В. Е. Графы и алгоритмы. Структуры данных. Модели вычислений / В. Е. Алексеев, В. А. Таланов. — Москва : Бином. Лаборатория знаний, 2012. — 320 с.

7. Ifenthaler, D. Informing learning design through analytics: Applying network graph analysis / D. Ifenthaler, D. Gibson, E. Dobozy // Australasian Journal of Educational Technology. — 2018. — Vol. 34 (2). — P. 117–132. https://doi.org/10.14742/ajet.3767

8. Асельдеров, З. М. Представление и восстановление графов / З. М. Асельдеров, Г. А. Донец. — Киев : Наукова думка, 2001. — 96 с.

9. Hypergraph-based data link layer scheduling for reliable packet delivery in wireless sensing and control networks with end-to-end delay constraints / Mao Yan, Kam-Yiu Lam, Song Han, Edward Chan // Information Sciences. — 2014. — Vol. 278. — P. 34–55. 10.1016/j.ins.2014.02.006

10. Grigor'yan, A. Introduction to Analysis on Graphs / Alexander Grigor'yan // Providence, Rhode Island : American Mathematical Society, 2018. — 150 p.

11. Anduo Wang. Ravel: A Database-Defined Network / Anduo Wang, Xueyuan Mei, Jason Croft [et al.] // In: Proc. Symposium on SDN Research. — 2016. — Art. 5. — P. 1–7. — URL: www.researchgate.net/publication/304918854_Ravel_A_Database-Defined_Network (accessed: 21.08.2021). https://doi.org/10.1145/2890955.2890970

12. The Comparison and Verification of Some Efficient Packet Capture and Processing Technologies / Jiaqian Li, Chengrong Wu, Jiawei Ye [et al.] // 2019 IEEE Intl. Conf. — 2019. — P. 967–973. — URL: www.ieeexplore.ieee.org/abstract/document/8890423 (accessed: 18.08.2021). 10.1109/DASC/PiCom/CBDCom/ CyberSciTech.2019.00177

13. Saavedra, M. Towards Large Scale Packet Capture and Network Flow Analysis on Hadoop / M. Z. N. L. Saavedra, W. Yu // In: Proc. 6th Int. Workshop on Computer Systems and Architectures. — 2018. — P. 186–189. — URL: www.researchgate.net/publication/329905189_Towards_Large_Scale_Packet_Capture_and_Network_ Flow_Analysis_on_Hadoop (accessed: 18.08.2021). 10.1109/CANDARW.2018.00043

14. Marton, J. Formalising openCypher Graph Queries in Relational Algebra / József Marton, Gábor Szárnyas, Dániel Varró // In: Proc. 21st European Conf. on Advances in Databases and Information Systems. — 2015. — Vol. 10509. — P. 53–68. 10.1007/978-3-319-66917-5_13

15. Graph Analytics using Vertica Relational Database / Alekh Jindal, Samuel Madden, Malu Castellanos, Meichun Hsu // IEEE Xplore. — 2015. — P. 1191–1200. — URL: https://arxiv.org/abs/1412.5263 (accessed: 18.08.2021).

Об авторах:

Галушка Василий Викторович, доцент кафедры «Вычислительные системы и информационная безопасность» ФГБОУ ВО «Донской государственный технический университет» (РФ, 344003, г. Ростов-на-Дону, пл. Гагарина, 1), кандидат технических наук, ORCID: http://orcid.org/0000-0003-2369-065X, galushkavv@yandex.ru.

Фатхи Денис Владимирович, доцент кафедры «Информационные технологии» ФГБОУ ВО «Донской государственный технический университет» (РФ, 344003, г. Ростов-на-Дону, пл. Гагарина, 1), кандидат технических наук, ORCID:https://orcid.org/0000-0003-1538-1363, Zmey2257@mail.ru.

Газизов Евгений Равильевич, доцент кафедры «Физика и математика» ФГБОУ ВО « Казанский государственный аграрный университет» (РФ, 420015, г. Казань, ул. К. Маркса, 65), кандидат физико-математических наук, ORCID: https://orcid.org/0000-0003-1538-1363, pim.kazgau@mail.ru.

Поступила в редакцию 26.07.2021
Поступила после рецензирования 09.08.2021
Принята к публикации 09.08.2021

Заявленный вклад соавторов:

В. В. Галушка — формирование основной идеи работы, постановка цели и задач исследования, разработка метода разделения множеств адресов. Д. В. Фатхи — разработка аппаратного комплекса захвата трафика, практическая реализация предложенных методов. Е. Р. Газизов — определение структуры базы данных и формирование реляционных операций пересечения множеств.

Все авторы прочитали и одобрили окончательный вариант рукописи.